

APPENDIX A
CLAIMS MARKED UP TO SHOW AMENDMENTS

Please amend the following claims:

9. (amended) A method for generating documents for display by a browser using components that react interactively on user input by executing instructions on a server, comprising the following steps for execution on the server upon a [page] document request:

assigning a unique identifier to at least one of the components; and
embedding the unique identifier into a generated [page] document.

12. (amended) The method of claim 11, wherein at least one of the components is contained on a [page] document template.

20. (amended) The method of claim 16, wherein the unique identifier is unique within all [pages] documents generated by a single server within a defined time period.

22. (amended) A computer running an application to develop and maintain applications using a web browser, comprising:

an editor operable within the web browser for inserting, deleting, and modifying components on document templates; and

a [page] document generator for processing document templates and for generating documents from the document templates that are understandable by the web browser.

24. (amended) A computer as in claim 23 wherein at least one of the components can react on subsequent [page] document requests containing user responses by executing selected instructions.

25. (amended) A computer as in claim 24, wherein the computer further comprises:
a store of component classes, each component class implementing one component kind;
and
a parser able to detect components marked on [page] document templates;

APPENDIX A
CLAIMS MARKED UP TO SHOW AMENDMENTS

wherein the [page] document generator works upon a [page] document request using component classes to generate browser code; and

wherein the editor is capable of showing a menu of components for insertion into the [page] document templates.

34. (amended) A method for generating a [page] document for display in a browser from a [page] document template containing components, comprising:

for each component denoted on the [page] document template, identifying a component class of the component; and

based on data contained in a request initiated by the browser storing a first object of the component class, the first object representing the component.

37. (amended) The method of claim 34, further comprising, for at least one component kind, for all components denoted on the [page] document template having said component kind; generating a unique identifier; assigning said unique identifier to said object, and embedding said unique identifier into the browser code.

39. (amended) The method of claim 34, wherein the [page] document template is parsed into a list of nodes, including text and component nodes, said method further comprising: determining if the current node is text or a component; if component, then calling a method for the component, comprising: evaluating the attributes of the component if necessary; identifying the component class associated with the component; and calling the constructor method of the component class, said constructor method generating browser code; if text, then generating the text; and repeating these steps for each node.

APPENDIX A
CLAIMS MARKED UP TO SHOW AMENDMENTS

44. (amended) The method of claim 34 wherein storing [an] the first object comprises creating a new object as necessary.

45. (amended) The method as in claim 34 wherein components are denoted on [page] document templates using tag syntax.

47. (amended) The method as in claim 36 wherein components are denoted on [page] [^]document templates using tag syntax, wherein the tag name identifies a component class.

51. (amended) A system for editing components on web document templates for use with a first software program including first instructions for generating a document request to obtain at least one generated document from a second software program and for displaying the generated document, the second software program capable of receiving and processing a document request and of transmitting first documents to the first software program in response to requests, said system comprising:

a plurality of components,

a plurality of document templates,

the second software program transmitting, while processing selected requests, second documents to the first software program that make the first software program display a user interface [to perform] for editing functions used for maintaining components on document templates,

a third software program used by the second software program while processing selected document requests, the third software program including third instructions for modifying document templates in order to perform said [an] editing functions.

58. (amended) The system of claim 56 further comprising instructions to allow the user to click on the generated [page] document to select items to perform edit functions on.

59. (amended) A software development system for dynamic web documents comprising:

APPENDIX A
CLAIMS MARKED UP TO SHOW AMENDMENTS

an editor program for editing dynamic web documents,
a document generator for generating generated documents from dynamic web documents,
the editor program comprising first instructions for requesting the document generator to process a dynamic web document leading to a generated document,
the editor program further comprising second instructions for displaying at least some information items contained on said generated document in a view which allows the user to select an item to which a modification function will be applied,
the editor program further comprising third instructions to modify the dynamic web document to perform said modification function.

89. (amended) A software development system as in claim 74 wherein at least one sixth component includes tenth instructions to display the sixth component, tenth [ninth] instructions being used to generate browser code for displaying [display] the sixth component during editing as well as during normal use of the component.

90. (amended) An editor for use with a web browser, the editor allowing the user to edit a document displayed by the browser, wherein clicking on said document displayed in the browser [window] initiates editing functions, and scripts contained in said document [staying] remain functional, the editor [comprising] including a first software program for execution within the browser and for processing the clicking on said document [providing the user interface of the editor].

97. (amended) A system for displaying dynamically generated documents in a data network coupling a server computer to a client computer, wherein the client computer has a first software program including first instructions for generating a document request to obtain at least one generated document from the server computer and for displaying the generated document, comprising:

a plurality of components for execution on the server computer, [at least one] including a first component including second program instructions to generate browser code and third

APPENDIX A
CLAIMS MARKED UP TO SHOW AMENDMENTS

program instructions for execution on the server which are initiated by the user interacting with the first component, and,

fourth program instructions on the server computer for, based on data contained in a document request initiated by the first software program on the client computer, generating generated documents for transfer to the client computer and display by the first software program, thereby calling second program instructions of components.

106. (amended) The system of claim 105, wherein fifth instructions call third instructions only if first component was contained on a page previously [displayed] transferred to the client.

124. (amended) The system of claim 114 containing at least one component wherein second program instructions are used to generate browser code for displaying [display] the component during editing and during normal use.